# EXHIBIT 43

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

| | |
|---|---|
| CISCO SYSTEMS, INC., Plaintiff, vs. ARISTA NETWORKS, INC., Defendant. | Case No. 5:14-cv-05344-BLF (PSG) |

VIDEOTAPED DEPOSITION OF GREG SATZ

March 23, 2016

HIGHLY CONFIDENTIAL - ATTORNEYS' EYES ONLY

VOLUME 1

Reported by
Brooke R. Bohr
CSR No. 753
Job No 2272380
Pages 1 - 168

VIDEOTAPED DEPOSITION OF GREG SATZ, taken at the instance of the Defendant, at the offices of TUCKER & ASSOCIATES, 605 W. Fort Street, in the City of Boise, State of Idaho, commencing at 10:10 a.m., on March 23, 2016, before Brooke R. Bohr, CSR, RPR, a Notary Public in and for the State of Idaho, pursuant to notice, and in accordance with the applicable Rules of Civil Procedure.

A P P E A R A N C E S

FOR PLAINTIFF
John M. Neukom, Esq.
QUINN EMAMUEL URQUHART & SULLIVAN LLP
50 California Street, 22nd Floor
San Francisco, CA 94111
(415) 875-6320
johnneukom@quinnemanuel.com

FOR DEFENDANT
Brian L. Ferrall, Esq.
KEKER & VAN NEST LLP
633 Battery Street
San Francisco, CA 94111
(415) 391-5400
bferrall@kvn.com



W I T N E S S



* * * * *

E X H I B I T S



* * * * *



BOISE, IDAHO
March 23, 2016, 10:10 a.m.

THE VIDEOGRAPHER: We are now on the record. Please note that the microphones are sensitive and may pick up whispering and private conversations. Please turn off all cell phones or place them away from the microphones as they can interfere with the deposition audio. Recording will continue until all parties agree to go off record.

My name is David Cromwell, representing Veritext. The date today is March 23, 2016, and the time is approximately 10:10 a.m. This deposition is being held at Tucker & Associates located at 605 West Fort Street, Boise, Idaho 83702, and is being taken by counsel for the defendant.

The caption of this case is Cisco Systems, Inc. v. Arista Networks, Inc. This case is filed in the United States District Court, Northern District of California, San Jose Division, Case No. 5:14-CV-05344-BLF PSG. The name of the witness is Greg Satz.

At this time, the attorneys present in



the room will identify themselves and the parties they represent.

MR. FERRALL: Brian Ferrall of Keker & Van Nest on behalf of Arista Networks.

MR. NEUKOM: John Neukom for the plaintiff.

THE COURT: Our court reporter, Brooke Bohr, representing Veritext, will swear in the witness, and we can proceed.

GREG SATZ,
produced as a witness at the instance of the Defendant, having been first duly sworn, was examined and testified as follows:

EXAMINATION

BY MR. FERRALL:

Q. Good morning, Mr. Satz. Can you please state your full name.

A. Greg Leonard Satz.

Q. Mr. Satz, you are not represented by counsel today; is that right?

A. Correct.

Q. Have you ever been deposed before?

A. I have.

Q. All right. So you know the basic

you were there?
A. Yes, that's fair to say.
Q. Did you have a role in determining how the command line interface would evolve?
MR. NEUKOM: Objection; vague.
Q. BY MR. FERRALL: Yeah. Strike that. Let me rephrase that.
A. I was going to ask you for a clarification.
Q. Did you have input into the command line interface?
A. Yeah. I was going to draw a distinction for the benefit of that. So the command line interface is a generic term, isn't descriptive enough, because there's the content of it and there's the mechanics of it. So for the purpose of answering, I had some responsibility for the mechanics of it; the content of it was distributed around the engineering organization in an effort to deliver services and products.
Q. Okay. Can you help me understand what -- explain a little more what the difference is between the mechanics and the content?
A. So everyone knows these days the term API, application programming interface. That's



probably the closest generic term that would describe the mechanics. We would provide a way other people could add a command, provide hooks back to code they would write. And so the -- what we call the parser itself would then take textual tokens, process them, and ultimately get some code executed. And we didn't have to know what code that was, that was for some other programmer to use the API, but it was my group's organizational responsibility or -- basically, Cisco had thousands of things swirling around and thousands of things constantly hitting it every few weeks. So you would just start picking things up and start doing them. And so I think I just picked the parser up and started doing it.
Q. So would the parser be part of the mechanics, what you refer to as the "mechanics"?
A. Yes, or the API. Basically, how you write a command.
Q. I see. All right. And then when you refer to the content of the command line interface, can you give me some examples of that?
A. Sure. The show process command or if you were using some sort of tunnel or security filter that you would block or permit or enhance



performance, a traffic prioritization. Somebody else would write that and then have the code that did that trick, whatever that trick was, and we didn't have anything to do with that aspect of it because that was somebody else's responsibility, somebody else's coding, another manager, another organization within engineering.
Q. So the commands themselves, would you consider that part of this API or is that distinct from it?
MR. NEUKOM: Objection; vague and compound.
Q. BY MR. FERRALL: Meaning the words of the command itself, is that --
A. Was content.
Q. That's content, not the API part?
A. Right.
Q. All right.
A. The code itself that did the parsing was ultimately what Terry Slattery and Rob Widmer and other folks redesigned.
Q. Let's go back a little bit to your pre-Cisco employment history. Can you summarize that for me, please.
A. I mean, how far back do we want to go?
Q. Well, just -- just in terms of the



places you've worked. If you could list where you worked after -- from college on.
A. So, technically, my first job out of college after my undergraduate degree was New York University Graduate School of Business; I then transferred to Menlo Park to SRI International; and from there to Stanford University; and from there to Cisco.
Q. Okay.
(Discussion off the record.)
(Exhibit 400 marked.)
Q. BY MR. FERRALL: So, Mr. Satz, as Mr. Neukom predicted, I presented to you a printout of what I believe is your LinkedIn profile; is that right?
A. Yes, it sure looks like me.
Q. All right. Okay. And is -- do you keep this relatively up to date?
A. Not very consistently, no.
Q. Okay. All right. I just -- I wanted to mark it for the sake of seeing if this -- it helps put dates on your various pre-Cisco employment.
A. Yeah, this is -- this is a good rendition.

"Stanford Ethertip/Gateway User and Configuration Guide."
A. Yeah.
Q. Had you ever seen this before?
A. I'm sure I have. I don't have a recollection of it, and I don't remember this date at all. This is a pretty late date.
Q. Do you know Glenn Truitt?
A. I do.
Q. What did he work on at Stanford?
A. I no longer remember. I do know that he had his hands in this software, but a lot of people did. Jeffrey Mobile, Benji Levy. This was -- this code was a lot of research work. And so if one of the graduate students felt there was an application they wanted to experiment with, this really was the beginning of what then became the multi-protocol router and Cisco's router. So -- oh, yeah, there's some really old -- really old stuff here.
Q. Did you become familiar with some of the commands from this device?
A. Yes.
Q. Yeah? How did you become familiar with it?



A. Well, we were users of these devices when I -- the state of the art back then, before there were all of these computers and laptops, is you used a basic terminal with RS232 into some device that converted the commands into network protocols and used that across the network to talk to mainframes. That was state of the art.
So on my desk at Stanford and at SRI was these computers that were just terminals. They -- all they did was take a capture of keypress and generate a character. And that character was shipped across the network somewhere. And the computer would get that character, do something with it, and ship you back the output.
So that's what a TIP was, terminal interface processor. It allowed you to take an RS232 terminal and sit on the network without talking to the computer directly. And I think Kirk was responsible for gluing the TIP and the Gateway software together, because they were two different software bases.
Q. So if I could ask you to turn to Page 6 of that Exhibit 36.
A. Okay.



Q. This was a -- begins a section called "privileged commands." Do you see that?
A. Um-hum. I do.
Q. And were you aware of a privileged mode in this -- in the TIP Gateway?
A. Sure.
Q. Explain what was the purpose of the privilege mode there.
A. It mimicked the TOPS-20 style of parsing, and it -- there were commands that people would use to just have the device do what it does day-to-day, and there were commands that administrators or users who needed to maintain the device in the network would use. And so privilege commands were the latter set, and TOPS-20 had a very similar model.
Q. And this document says in the -- I guess in the second sentence, it's -- or I'll read the first sentence also: There's a second set of commands available to the Ethertip user. The two command levels are disjoint. That is, the privileged mode is not a superset of the normal mode.
Do you see that?
A. Um-hum.



Q. So what did you understand to be the purpose of the normal mode, then, as opposed to the privileged mode?
A. Day-to-day users don't need privileged mode. They go in, they make their connections, they do what they do to get their work done, and that's the extent of their relationship to the software.
The people who administer the device and who might need to add a new feature or upgrade the software would have to use privileged mode. And it is a complete separate set of functions.
And in particular for the programmers, they -- you know, they made a mistake, and they've got to go figure out why something is not working, especially for research work.
Q. I want to ask about some of the commands that follow here on Page 6. "Access.lists," I see under 3.1.
A. Um-hum.
Q. Was that a command you were familiar with?
A. That's a very common and important command.
Q. What is an access.list command?

A. It gave the device the ability to decide what data flows it would allow through or prevent, and/or connections that people could make to the box. So if, for example, your department wasn't allowed to use this particular box, we could create an access.list so you couldn't use it.
Q. Was -- to your knowledge, was access.list used in any other operating systems or softwares, software?
MR. NEUKOM: Objection; foundation.
THE WITNESS: I can't say I've ever seen access.list before this application.
Q. BY MR. FERRALL: The next command here is -- it says "arp.table." What is that, do you know?
A. Yes, address resolution protocol. So that was the mechanism that computers used to discover each other's -- I'm going to get really boring here -- 48-bit ethernet address and match it to their 32-bit IP address.
Q. And was the address resolution protocol something that was known outside of the Stanford network context?
A. Oh, yes, it was a standard.



MR. NEUKOM: Objection; lack of foundation, calls for speculation.
Q. BY MR. FERRALL: Now, don't worry, I'm not going to go through every command in here.
A. You'll need coffee.
MR. NEUKOM: And I think, actually, Brian, I didn't -- you said a little while back that the next -- just to make sure the transcript is clean, after discussing access.list, you said the next one listed is arp.table.
MR. FERRALL: That's -- you're right. That's not correct. That was -- I skipped one.
MR. NEUKOM: Okay.
MR. FERRALL: Thank you.
Q. BY MR. FERRALL: Were you familiar with show commands used in the Stanford TIP?
A. Yeah. I didn't do a lot of work with the TIP, so I can't say I have a great familiarity with this version of the software.
Q. Okay.
A. By the time I spent time with the software, it had been rewritten and the parser, which is the interesting part for this discussion, had been changed. So a lot of these commands are almost like looking at them new again. There's



just some here I don't -- I have no recollection of.
Q. Had you ever heard of or used show commands in any context before you went to Cisco?
A. Every computer has show commands. I mean every operating system had used the word "show" as a way to convey internal information outward.
Q. What about banner, which, by the way, I see at the bottom of Page 8 of Exhibit 36. But my question is more general, which is were you aware of a banner command before you went to Cisco?
A. I don't remember. I had used, by then, anywhere from 15 to 20 different operating systems. And so I -- banner doesn't stand out as anything.
Q. If I could ask you to look at Page 13 of this exhibit, Exhibit 36. Do you see on that page there are a number of commands that have in brackets the word "no" before the command?
A. Um-hum. I do.
Q. Do you know what that means?
A. It is an optional keyword.
Q. And what does it do? What function



does it serve?
A. Excuse me. Optional keywords just allow you to include them or not include them. So you -- I don't know if you're asking me what it means in relationship to each command or the generic optional keyword.
Q. Well --
A. There's layers.
Q. Okay. Fair enough.
Is there an overall purpose of -- if you wanted -- if you decided to include "no" as an option before these commands, is there a generic -- general way of describing what that would do?
MR. NEUKOM: Objection; vague and compound.
THE WITNESS: It is a negation in this particular usage. This, again, gets back to the mechanics or the API versus the content. So the confusion is I'm going to -- as a somewhat of a programmer or someone who has worked at the building of this, I'm more coming from a here is the mechanical word. An optional keyword is an optional keyword. What it actually means is going to be interpreted by the code that is written. So in this particular case, the "no" in front would

did, while it had the same capability, was more robust, had a higher performance capability. Because as the networks evolved, you needed to be able to push data faster. And Stanford's code was basic. It was there to just move the data, not move it with the requirements that the next few years dictated. And a lot of what Kirk did was to create high-speed interfaces, and that's what Wellfleet showed up to compete on was could they go faster than Cisco. And it created an arms race, as it were. Who could go faster.

Q. Now, you mentioned IETF, and I think earlier today you mentioned RFCs. Can you tell me what an RFC is?

A. Request for comments.

Q. And what's the purpose of a request for comment?

A. To create a protocol definition or solution and to publish it as a request for comments in an effort to move it forward as a proposed solution and a trial solution and then a committed solution, as the solution progressed through a community and an implementation and a trial and then some feedback. So it was an engineering group. Their goal was to deliver



something working. Companies would try to use it as -- to competitive advantages. But the standards body existed to create a level playing field.

Q. And did you have a view at the time as to the importance of publishing technology through RFCs? Well, let me strike that. That was a garbled question.

In your experience at Cisco in the early years, was the sharing of technology through RFCs important to Cisco?

MR. NEUKOM: Objection; vague, compound, and lack of foundation.

THE WITNESS: Back then it wasn't clear how successful Cisco would be and/or whether we might maintain or keep a competitive advantage. So there really was a series of tradeoffs in the decision to create an RFC and make it a community effort or to create a proprietary solution and then decide whether to make it an RFC later. Most of the times, it was to make the customers happy. If the customers wanted something documented, we would typically figure out how to comply with that. Later, as the company got larger and I wasn't involved, managing the IETF and RFC process



evolved. And I can't speak to that as much.

Q. BY MR. FERRALL: Okay. But --

A. But managing that was important.

Q. And just by way of example, you mentioned IGRP.

A. Um-hum.

Q. And that was a technology that Cisco chose to keep proprietary, right?

A. Yes.

Q. All right. And there were other technologies that Cisco was involved in developing, like BGP, for example?

A. Right.

Q. And that Cisco chose to publish RFCs about, right?

A. Well, Cisco didn't publish the RFCs. Cisco -- a person like Kirk might be a part of the team that developed BGP and then Kirk would have his name on it with a Cisco title, but it wasn't Cisco, it was actually Kirk. And the RFC itself is an open document. So just to make that distinction.

If there was a protocol that showed up from the IETF, Cisco was typically involved.

Q. And what was your involvement in



IETF -- in IETF? Did you --

A. I would go to the meetings and attend various functions and decide, based on the software responsibility I had, to participate in different standards or not.

MR. FERRALL: Let's mark this as the next exhibit.

(Exhibit 403 marked.)

THE WITNESS: More ancient history.

Q. BY MR. FERRALL: Yeah. So I've marked as Exhibit 403 what I think is an IETF RFC for a simple network management protocol, SNMP. Do you recognize this, Mr. Satz?

A. I do.

Q. Did you have involvement in the SNMP RFC?

A. I did.

Q. What was that involvement?

A. I was just part of the working group that went through the process of deciding what would be done as a solution to network management. And SNMP was the output.

Q. Do you remember when this SNMP working group began to discuss this solution?

A. Probably a couple years before this

document, at least a year.
Q. And do you remember any particular parts that you contributed, specifically?
A. I think I did an RFC for a MIB for CLNS, another protocol stack that since disappeared.
Q. Was there a -- have you ever heard of the term "SNMP server"?
A. Oh, the command line, parsed for the -- yeah -- configuration? Um-hum. Yes, I created that.
Q. What's -- is there such a thing as an SNMP server, or what does that term mean?
A. Wow.
MR. NEUKOM: Objection; lack of foundation, calls for opinion testimony.
THE WITNESS: I think all of that code is gone now. The SNMP server was the way to tell the router software that it was to be an SNMP -- it was to start the SNMP protocol. So it would then begin to listen to and process SNMP packets. And it was probably one of the first commands implemented as part of this RFC to implement it and create an SNMP protocol within the Cisco software.



called an SNMP community.
Do you see that?
A. Yes.
Q. Is that consistent with your definition of SNMP community that you just described?
A. Yeah. It's more mind-numbing when you see it in words.
Q. I couldn't agree more.
A. Yeah. It turns out a lot of these things are written to be really obtuse. They are not intended to be obtuse, but they have a structure to them that when you turn it into English or a simple picture it takes a lot of this out. They tried to make a more generic mathematical underpinning to a mapping that added a level of complexity that just ultimately wasn't necessary. But they were trying to be very flexible.
Q. Okay. But this notion of community as described in the Exhibit 403 is the same as the community that you understood when you --
A. I made the implementation simpler because of adding a whole layer. The idea, if I can remember any of this craziness, is that you would have a table of -- no different than a



MR. NEUKOM: And, Brian, I rescind my prior objection. Pardon me.
THE WITNESS: Hey, just because I write it, doesn't mean I'm the expert.
MR. FERRALL: You can't -- you can't rescind. No rescinding objections, Mr. Neukom.
Q. BY MR. FERRALL: What's -- what's the notion of community in the context of SNMP?
A. After a while, you start running out of words, so you pick one that tries to create a sense of purpose. And so "community" was an attempt to describe a collection of users who would have a specific purpose with respect to using the protocol. It was nothing more than an authorization or an access. A password, as it were.
Q. So if you look at Page 7 of this Exhibit 403.
MR. NEUKOM: Sorry. Which page are we on?
MR. FERRALL: Page 7.
Q. BY MR. FERRALL: If you see under Section 3.2.5, Definition of Administrative Relationships, and then the second paragraph there says, quote, appearing of an SNMP agent with some arbitrary set of SNMP application entities is



database in today's language -- and you could be able pull out individual things. And so they wanted to be able to map authorizations to individual entries in the database. And the implementation I did was to make it an all or nothing. Because if somebody wanted that level of specificity they'd ask for it and then we'd go back and put all that crazy complexity into the code. But just because the standard made it that flexible we weren't going to go that far. It was an engineering choice and cost benefit.
Yeah, I don't know if you've ever heard of Vint Cerf?
Q. Sure.
A. So one of the more inspiring aspects of this work, we had three different protocols compete to be the network management RFC, and so there was just three groups of engineers that were not happy, or wanted their choice. And I watched Vint come in and broker a -- mediate, and I had never seen that kind of mediation happen before, let alone difficult engineers. And so it was a very inspiring time to watch somebody. And then so, you know, Vint was the author of a lot of the TCP/IP protocols. So people respected him and

were willing to listen to his know. So I -- that's a really good story about Vint.

Q. Well, on this issue of SNMP community, do you remember how this discussion came about in the context of the IETF working group?

A. What happened with this protocol is these four people went off and created this, brought it back to the IETF and said we should use this. And it was actually done at the behest of the mediation result by Vint to say, okay, we're going to take all of these ideas that you have for all of your different proposals and we're going to bring it down to this. So they went and did this. So it was really done in a small group of these four, maybe a half-dozen people, published this document and then they brought it back to the IETF to ratify or get the feedback loop that the IETF is.

So the communities were intended to be a very flexible generic solution to an access mechanism as they wrote it.

Q. Okay. And so you -- you implemented a simpler version --

A. A subset.

Q. -- of the community that these four



authors had proposed.

A. Yeah. Exactly. And that was in this '89 timeframe. Since then the code has been -- that I wrote has been thrown out and they bought a third party, I think from Mr. Case. I think Cisco went and bought his software and just got a whole bunch of new features instead of writing it themselves and moved on. And I have no idea what those parse commands look like. I don't even think I've -- I still run a Cisco router, and I don't think I've enabled SNMP.

Q. So in implementing, for example, the SNMP server community function, were you responsible either directly or indirectly for implementing the functional code?

A. I was.

Q. All right. And was that directly? Were you actually writing that --

A. I wrote the code.

Q. You wrote the code?

A. The first version, yes.

Q. And for that function, do you have any sense of how long it took you to write that code?

A. No. Because it was all part of a much larger set of functionality, and so to pull out



that one little line of parsing, no. I mean overall the task was probably two to three months to do the full SNMP stack.

Q. Okay. And how long did it take you to come up with the names for the commands for the -- for SNMP functionality?

A. 15 seconds, conceptually, five seconds. I mean, this is the name, type it in, move on.

Q. And to be clear, though, the original parser on the first generation of Cisco products, that was already written by the time you joined; is that right?

A. Well, the EtherTIP-style parser, to use this document as a basis, wasn't a parser in a sense that it had a common code base. It was just individual programming statements to consume tokens. So the distinction being there's a bunch of subtertians (sp) whose job it is to do something versus individual lines of code scattered thousands of places that consumed tokens that parse. So that was the EtherTIP style and then Terry Slattery and Widmer put it together in a common set of code and created data structures and a programming interface and documentation on how to use it.



So by the time -- I don't remember where SNMP falls within that. And I can't remember when I typed it in, if I was typing it into the old style or Slattery had already done his work. Do you have a date for Slattery's --

Q. Yeah. And that's what I was just looking for. I don't --

A. Yeah, because I think I -- I think these commands were before Slattery's work, because I was a manager by then when I got Slattery to redo the parser.

Q. Right. And I will want to get to that. I've got to dig documents out. But at 1990, 1991, does that sound about right --

A. Yeah.

Q. -- for Terry Slattery's work?

A. Yeah. This was the late '80s, yeah.

Q. Okay. And while we're there, am I right that you were the person who hired Terry, right?

A. I did.

Q. Okay. All right. We'll come back to that in a little bit.

You mentioned an RFC for a MIB?

A. Um-hum.

service. And so I called them a server, like you would on an operating system running as a separate process. So it was just a distinction I happened to use just from where I had come from.

Q. In the Unix context, what -- how was that manifest itself, or what's an example of that in the Unix context?

A. Well, Unix is what is on these phones. It is Linux. It is just the next generation of it. And so it is any arbitrary process running in the background that people might call a demon that provides a service. I mean you -- your -- you go to the web, you're talking to a web server. It just happens to be a dash in the configuration language.

Q. Right.

A. Maybe that helps with the modern analysis in comparison, as opposed to a routing protocol or a switching engine or a link layer, like an ARP. I mean, there's all these different components.

Q. Okay. I think we can put that aside.

So are you familiar with the terminal monitor command?

A. Yes.



Q. And do you know the origins of that?

A. I think I wrote it.

Q. Okay. What function does the terminal monitor command invoke?

A. I now use it without thinking. So the ability to figure out what's happening in a piece of software requires some diagnostics. And so we created a lot of debug commands that would print out the debugging. The debugging typically only went to the console which, in the good old big iron hardware, wasn't a bitmap display, but just an RS-232 port, and usually it was hooked to a good old-fashioned terminal in today's perspective. So the stream of debug diagnostic messages would come out this console port and if you're sitting at home, trying to connect in and do some debugging, it couldn't get there from here. The data was going over to your office in some terminal and the only way to look at it was to attach a stream back to where you were, and that was what monitor did. It said send me anything that came out on the console to my virtual terminal connection.

Now all laptops have bitmap displays and fancy graphics and no one even says RS-232



anymore.

Q. So am I right, then, that "monitor" in the command terminal monitor refers to monitoring the bug diagnostics?

A. It is actually monitoring anything that gets printed to the console port.

Q. Okay.

A. Which the important stuff was the diagnostics. It is the old world screen sharing of today.

Q. And did you write the code to implement that feature?

A. Yes.

Q. When did you do that, approximately?

A. Wherever it shows up in the manuals.

Q. Early, early years?

A. Yeah, because we needed that to help improve our proficiency to debug so we didn't have to be at the office.

Q. And how did you come upon the selection of the command terminal monitor for that?

A. The same expediency I did all of them: Monitor, sounds good, next.

Q. Okay.

A. Yeah. Unless Kirk didn't like my



choice, I think it was just whatever that struck me as a -- as what it did as I could perceive it from the point of view at the time.

Q. Have you ever heard of the term -- well, strike that.

Have you ever heard of people in your field characterizing a command as a "generic command"?

A. Yes.

Q. What does that mean to you?

A. Like "show." It is everywhere.

Q. And how would you contrast that concept of a generic command like show versus a non-generic command?

A. Its applicability to many different aspects or areas. So to -- I mean to use the word "generic" is not really clear, but it's -- probably if you look at it in the hierarchy sense, the top note is pretty generic. And depending on how many commands under it -- so relationship to the other commands around it or below it, if it is the root of a very deep tree, it's going to be more generic than if it's just one layer deep. And its applicability therefore expands out.

Q. So if I understand you, then, at a high

Q. Yeah.
A. I wasn't hands-on at that point.
Q. Okay.
A. There are generations of people who went on to do that stuff.
Q. So, Mr. Yeager -- you're not Mr. Yeager. I was looking at. . .
Mr. Satz, do you own Cisco stock anymore?
A. No. Well, I probably have some shares in a retirement account, I think. So, yeah, I probably have a few, but not a lot.
Q. Do you own any Arista stock?
A. No. I have stayed out of the technology buying. I had enough risk in that.
Q. That's a wise move for the past six months, right?
A. Well, pick your time.
Q. Right. Okay.
Did you -- have you spoken to anyone in-house at Cisco, employee at Cisco, about this case?
A. Kirk. After Morgridge, I had dinner with him, and then I had dinner with him again a few weeks ago. Both times it came up; both times



he said, "After your deposition we'll talk. I'm not going to talk to you before." And I said, "Fine with me."
I'm trying to remember if there was anything interesting. He was pretty good -- my wife was commenting on it, "Yeah, he really wanted to tell you stuff, but he really kept his mouth shut." Yeah, that was -- he's smart.
I'll have to -- let's let that sit. We're getting to the end of the day. It's not working as well.
Q. Anyone other than Mr. Lougheed?
A. E-mail with Terry Yeager. I think I touched Tony Li on Facebook. He just said "yep" in his way. I think that's about it.
MR. FERRALL: Okay. Well, thank you for your time, Mr. Satz. I have no further questions.
THE WITNESS: Wow, that's an abrupt ending. Thank you.
MR. NEUKOM: I have a few.
THE WITNESS: Please.

EXAMINATION

BY MR. NEUKOM:
Q. A little bit earlier today, I think you



testified that you don't remember any instances, any historical instances, of Cisco using the word "proprietary" with respect to the CLI command set. Do I have that right?
A. Yes.
Q. Okay. So I just have one question which goes the other way.
Based on your experiences at Cisco, do you remember any instance in which Cisco made an announcement or made a statement that its CLI command set was -- it was okay if competitors wanted to use it?
A. Never heard that either.
Q. Okay. And you -- I think you smiled a little bit when you said you never heard that. Why?
A. No, it's a brilliant question. I appreciated it.
The only thing I can say in my recollection of usage is because -- I wouldn't call it a universal language, but it was the Cisco language, that there was a big effort to script Cisco. And outside of SNMP, there was no real remote control interface. So if you as a network operator wanted to toggle an interface up



or down or change some performance parameters, there was a whole push in the '90s to connect to the router, type -- robotically type commands at it.
And so there were scripting interfaces or scripting attachments that, you know, as engineers we're, like, okay, we could see your need to do that. Great. And we don't have a better solution than that and that's the way to control it in any way you want. We don't have to write SNMP every time you need one new thing. So it became the control language or the scripting language. So I wouldn't say we said it was open, but we knew and needed that customers have the ability to meet their needs controlling it.
Q. Sure. But to the best of your memory --
A. No.
Q. -- you never remember Cisco ever making a statement publicly or privately --
A. No.
Q. -- that it was okay for any competitor to be using this --
A. No. Not competitors, no.
Q. -- CLI command set?

Exhibit 405 is a one-page document marked CSI-CLI-00746924.

Exhibit 406 begins CSI-CLI-01828732, and for this document I'll read the last number because I think we're all unclear whether it is one versus multiple documents. This ends with Bates stamp CSI-CLI-01828783.

Exhibit 407 begins Bates stamp CSI-CLI-01295215.

And Exhibit 408 begins CSI-CLI-01295181.

MR. NEUKOM: Thanks all.

MR. FERRALL: Agreed. Thank you.

(The deposition concluded at 3:31 p.m.)

-oo0oo-



V E R I F I C A T I O N

I declare under penalty of perjury under the laws that the foregoing is true and correct.

Executed on _____ , 20___, at _____, _____.

_____

WITNESS SIGNATURE



R E P O R T E R ' S C E R T I F I C A T E

I, BROOKE R. BOHR, a Notary Public in and for the State of Idaho, do hereby certify:

That prior to being examined, the witness named in the foregoing deposition was by me duly sworn to testify the truth, the whole truth, and nothing but the truth;

That said deposition was taken down by me in shorthand at the time and place therein named and thereafter reduced into typewriting under my direction, and that the foregoing transcript contains a full, true, and verbatim record of the said deposition.

I further certify that I have no interest in the event of the action.

WITNESS my hand and seal March 30, 2016.

<%signature%>
Brooke R. Bohr
CSR No. 753